Time Series Forecasting of Confirmed Covid Cases in India

Abstract

The COVID-19 virus is negatively impacting the health care services of almost more than 90% countries in the world.The virus is spreading at a community level. Therefore, it becomes really important to analyse the possible impact of COVID-19 and forecast how it will behave in the coming days.We do comparative analysis of the using different machine learning techniques to predict the future Confirmed cases in this Covid19 pandemic. We use open source APIs to collect dataset required for time series analysis.

1. Introduction

Time series forecasting is applied on non-stationary observed data, to predict future values. Machine learning techniques are used to build models which learn the behaviour of the observed time series data. The models are then used to predict the future values.

2.Dataset

The open source project https://github.com/pomber/covid19 provides the covid time series data in a json file.
The file is accessed using https://pomber.github.io/covid19/timeseries.json. With the requests package in python, we collected the json file and placed the data of India into a csv file.

3. Preprocessing Data

Data collected from different online sources is integrated and stored in a csv file. The content is then cleaned by replacing the null values in the place of deaths or cured or confirmed with zero, checked for the presence of floating values and replaced with the integral parts of the values and checked for the presence of negative numerical values and raised an exception in case they are found. The collected data gave the accumulated cases information for the dates. We preprocessed the data to obtain the cases received on each day. Min max scaling is applied to the data.

4. Data visualization and analysis

1. Time Series visualization The behaviour of the time series collected and predicted are compared using the matplotlib library in python. By this visualization of the data, drawbacks and the reason behind the drawbacks of the model were analysed.
2. Map visualization The heat map is visualized using plotly package and pycountry package is used to visualize the change in heat intensity with the progression of days in python.
3. Data Analysis The analysis of the obtained data is done using the read_csv() and describe() functions in python.

5. Models Applied

5.1 ARIMA model

Used stats models package(https://github.com/statsmodels/statsmodels) for:

1. Plotting
   (a) Rolling mean
   (b) Standard deviation
   (c) ACF and PACF
2. Dickey fuller test
3. Arima modelling

To study the stationarity of the time series data, the rolling mean and standard deviation are plotted: The seasonal and residual properties are obtained by decomposing the dataset. The Dickey Fuller test result gave the p-value is above the critical value at even 5%. So it is assumed that the series has a unit root. So, differencing the series gave proper stationary series.

5.2 Machine Learning Models

1. SVM: Polynomial Kernel of degree as 4 is taken and the following hyperparameters are tuned:
   (a) gamma
   (b) epsilon
   (c) C
2. Linear Regression: PolynomialFeatures with degree as 2 is taken for preprocessing.

5.3 Multiple Layer Perceptron

Relu is taken as the activation function. Used lbfgs optimizer. The number of iterations is tuned.

5.4 Long Short Term Memory

Applied statefull LSTM model as the intial trails of using stateless model gave bad results. Mean Squared error is taken as the loss function. Adam optimizer is applied for better performance. The sequence length of the data for training and testing during learning is tuned. The hidden layer size and the learning rate are tuned. The epochs of training is tuned by observing the graph of loss function for training and testing data.

6. Evaluation Metrics and Results

The metric taken to evaluate the models and compare their performance is Mean Absolute Percentage Error(MAPE). Predictions are obtained for the last 15 days of the time series and evaluation is done using MAPE metric for all the models.
Hence a common base is maintained to compare the performance of the models accurately.
The MAPE results obtained for the models are as follows:

1. ARIMA: 5.99%
2. LSTM: 9.56%
3. Machine Learning Techniques
   (a) SVM: 16.05%
   (b) Linear Regression: 59.30%
4. Multiple Layer Perceptron: 59.46%

7. Conclusion

It has been observed that the ARIMA gave the best results closely followed by the LSTM model. From the visualization of the predictions, it was observed that both ARIMA and LSTM learned the zonal behaviour and the trend and gave closer predictions.Though MLP implements neural networks, it performed the least. It was not able to learn the behaviour of the time series properly because of the very few layers of neurons. SVM gave better performance compared to Linear Regression as it was able to learn the trend of the series, whereas Linear Regression only learned that the series was on rise and tried to adjust the slope of the predictor line accordingly.

Appendix

Acknowledgements

We thank the constant efforts of our professor, Dr. Bhaskar Biwas, for his constant motivation and guidance throughout this project. It was only because of his efforts that we were able to understand and convey the ideas and work on this project report.